

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
ЗАПОРІЗЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

МАТЕМАТИЧНИЙ ФАКУЛЬТЕТ

Кафедра програмної інженерії

**КВАЛІФІКАЦІЙНА РОБОТА БАКАЛАВРА**

на тему: «РЕАЛІЗАЦІЯ БІЗНЕС-АНАЛІЗУ ЗАСОБАМИ  
PYTHON У КОРПОРАТИВНІЙ ІНФОРМАЦІЙНІЙ  
СИСТЕМІ ТЕКСТИЛЬНОГО ПІДПРИЄМСТВА»

Виконав: студент 4 курсу, групи 6.1210-1пi  
спеціальності 121 інженерія програмного забезпечення  
(шифр і назва спеціальності)

освітньої програми програмна інженерія  
(назва освітньої програми)

О.В. Коростильов

(ініціали та прізвище)

Керівник доцент кафедри програмної інженерії,  
доцент, к.ф.-м.н. Горбенко В.І.  
(посада, вчене звання, науковий ступінь, прізвище та ініціали)

Рецензент доцент кафедри комп'ютерних наук,  
доцент, к.т.н. Борю С.Ю.  
(посада, вчене звання, науковий ступінь, прізвище та ініціали)

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ**  
**ЗАПОРІЗЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ**

Факультет математичний

Кафедра програмної інженерії

Рівень вищої освіти бакалавр

Спеціальність 121 інженерія програмного забезпечення

(шифр і назва)

Освітня програма програмна інженерія

**ЗАТВЕРДЖУЮ**

Завідувач кафедри програмної інженерії, к.ф.-м.н., доцент

\_\_\_\_\_ Лісняк А.О.  
(підпис)

“ \_\_\_\_\_ ” \_\_\_\_\_ 2023 р.

**ЗАВДАННЯ**  
**НА КВАЛІФІКАЦІЙНУ РОБОТУ СТУДЕНТОВІ**

Коростильову Олегу Владиславовичу

(прізвище, ім'я та по-батькові)

1. Тема роботи Реалізація бізнес-аналізу засобами Python у корпоративній інформаційній системі текстильного підприємства

керівник роботи Горбенко Віталій Іванович, к.ф.-м.н, доцент

(прізвище, ім'я та по-батькові, науковий ступінь, вчене звання)

затверджені наказом ЗНУ від « 21 » грудня 2023 року № 2180-с

2. Строк подання студентом роботи 03.06.2024 р.

3. Вихідні дані до роботи 1. Постановка задачі.

2. Перелік літератури.

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити)

1. Постановка задачі.

2. Основні теоретичні відомості.

3. Розробка архітектури інформаційної системи.

4. Реалізація програмних модулів.

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень) \_\_\_\_\_

презентація за темою доповіді

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		Завдання видав	Завдання прийняв

7. Дата видачі завдання 25.12.2023 р.

## КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
1.	Розробка плану роботи.	09.01.2024	
2.	Збір вихідних даних.	23.01.2024	
3.	Обробка методичних та теоретичних джерел.	13.02.2024	
4.	Розробка першого та другого розділу.	29.03.2024	
5.	Розробка третього розділу.	13.05.2024	
6.	Оформлення та нормоконтроль кваліфікаційної роботи бакалавра.	27.05.2024	
7.	Захист кваліфікаційної роботи.	17.06.2024	

Студент \_\_\_\_\_  
(підпис)

О.В. Коростильов \_\_\_\_\_  
(ініціали та прізвище)

Керівник роботи \_\_\_\_\_  
(підпис)

В.І. Горбенко \_\_\_\_\_  
(ініціали та прізвище)

### Нормоконтроль пройдено

Нормоконтролер \_\_\_\_\_  
(підпис)

А.В. Столярова \_\_\_\_\_  
(ініціали та прізвище)

## РЕФЕРАТ

Кваліфікаційна робота бакалавра «Реалізація бізнес-аналізу засобами Python у корпоративній інформаційній системі текстильного підприємства»: 41 с., 25 рис., 17 джерел.

АНАЛІТИКА ДАНИХ, ВЕБІНТЕРФЕЙС, ВІЗУАЛІЗАЦІЯ ДАНИХ, ЗВІТУВАННЯ, МОДЕЛЬ ПРОГНОЗУВАННЯ.

Об'єкт дослідження – процес реалізації інтегрованої системи прогнозування браку на основі моделі машинного навчання, що містить вебінтерфейс для введення даних, аналіз даних та візуалізацію результатів.

Предмет дослідження – інтеграція моделі машинного навчання в інформаційну систему, розробка функціоналу для збору і обробки звітності, візуалізація статистичних даних, керування доступом до ресурсів між користувачами.

Мета роботи: розробити інтегровану систему прогнозування браку на основі моделі машинного навчання для покращення управління якістю продукції та оптимізації виробничих процесів.

Методи дослідження – аналіз предметної області та функціональності застосунку, експерименти щодо реалізації функціональності застосунку.

Виконано аналіз предметної області, огляд інструментів для розробки інформаційних систем та аналіз використаних технологій. Розроблено архітектуру застосунку. Спроектовано базу даних і представлено діаграму її структури.

Розроблено серверну та клієнтську частину застосунку. Серверну частину реалізовано засобами мови Python та фреймворку Flask, клієнтську – засобами мови TypeScript та фреймворку React.

## SUMMARY

Bachelor's qualifying paper "Implementation of Business Analysis in The Enterprise Information System of a Textile Factory by Python": 41 pages, 25 figures, 17 references.

DATA ANALYTICS, WEB INTERFACE, DATA VISUALIZATION, REPORTING, PREDICTION MODEL.

The object of study is the process of implementing an integrated reject prediction system based on a machine learning model that includes a web interface for data entry, data analysis, and visualization of results.

Subject of study: integration of a machine learning model into an information system, development of functionality for collecting and processing reports, visualization of statistical data, and management of access to resources between users.

The aim of the study is to develop an integrated reject prediction system based on a machine learning model to improve product quality management and optimize production processes.

The methods of research are analysis of the subject area and functionality of the application, experiments on the implementation of the application functionality.

The analysis of the subject area, the review of tools for developing information systems, and the analysis of the technologies used were performed. The application architecture is developed. The database is designed and a diagram of its structure is presented.

The server and client parts of the application were developed. The server side was implemented using Python and the Flask framework, and the client side was implemented using TypeScript and the React framework.

## ЗМІСТ

Завдання на кваліфікаційну роботу.....	2
Реферат .....	4
Summary .....	5
Вступ.....	8
1 Використання бізнес аналізу на текстильному підприємстві .....	10
1.1 Задачі бізнес аналізу .....	10
1.2 Бізнес аналіз на текстильному підприємстві.....	11
1.3 Задачі для створення застосунку бізнес-аналізу.....	12
1.4 Аналіз популярних програм/систем для бізнес аналітики на підприємствах.....	13
2 Аналіз та проектування системи .....	16
2.1 Аналіз вимог .....	16
2.1.1 Опис цілей проєкту.....	16
2.1.2 Функціональні вимоги.....	16
2.1.3 Нефункціональні вимоги.....	17
2.1.4 Цільова аудиторія .....	18
2.2 Аналіз інструментів .....	18
2.2.1 Мова програмування Python .....	18
2.2.2 Фреймворк Flask .....	18
2.2.3 Реляційні СУБД.....	19
2.2.4 Мова програмування TypeScript.....	19
2.2.5 Фреймворк React .....	19
2.3 Архітектурне проектування .....	20
2.4 Проектування бази даних .....	21
3 Розробка системи .....	24
3.1 Розробка серверної частини .....	24
3.1.1 Автентифікація та авторизація .....	24

3.1.2 Обробка надання доступу на основі ролей .....	24
3.1.3 Попередньо визначені ролі .....	25
3.1.4 Взаємодія з базою даних .....	26
3.1.5 Прогнозування.....	28
3.1.6 Збір звітності з виробництва.....	28
3.1.7 Візуалізація даних.....	28
3.1.8 Адміністрування .....	29
3.2 Розробка клієнтської частини.....	29
3.2.1 Загальна структура клієнту.....	29
3.2.2 Опис наявних сторінок.....	32
3.2.3 Керування доступом до ресурсів.....	37
Висновки .....	38
Перелік посилань.....	40

## ВСТУП

Збільшення попиту на продукти текстильного виробництва змушує виробників задумуватись про оптимізацію виробничих процесів. Одним з популярних рішень даної проблеми є впровадження інформаційних систем.

Переваги інформаційних систем:

- оптимізація та автоматизація процесів, що збільшує швидкість та ефективність виробництва;
- економія ресурсів та спрощення управління витратами;
- забезпечення точнішої аналітики для прийняття обґрунтованих рішень;
- можливість швидкого реагування на зміни ринкових умов, що підвищує конкурентоспроможність підприємств.

Для текстильних підприємств розробляються різноманітні інформаційні системи що враховують специфічні вимоги кожного підприємства. Однак більшість цих рішень є пропрієтарними, а малі чи середні виробники не завжди можуть дозволити собі їх розробку та впровадження. Це обумовлює актуальність даного проєкту, який спрямований на задоволення попиту на доступні та ефективні інформаційні системи.

Метою кваліфікаційної роботи є розробка інтегрованої системи прогнозування браку на основі моделі машинного навчання для покращення управління якістю продукції та оптимізації виробничих процесів.

Досягнення даної мети потребує виконання наступних кроків:

- аналіз предметної області, реальних потреб підприємств та інструментів для реалізації продукту;
- підбір даних для навчання та підготовка моделі машинного навчання для прогнозування браку;
- розробка та документування серверної частини додатку (включно зі



структурою бази даних);

- розробка та документування клієнтської частини додатку;
- підготовка до розгортання на виробництві.

# 1 ВИКОРИСТАННЯ БІЗНЕС АНАЛІЗУ НА ТЕКСТИЛЬНОМУ ПІДПРИЄМСТВІ

## 1.1 Задачі бізнес аналізу

Бізнес-аналіз є стратегічним процесом, спрямованим на аналіз бізнес-процесів та виявлення можливостей для покращення в діяльності підприємства [17]. Він включає в себе систематичний аналіз внутрішніх та зовнішніх факторів, що впливають на організацію, з метою забезпечення ефективного управління ресурсами та досягнення стратегічних цілей компанії.

Роль бізнес-аналізу на підприємстві полягає у визначенні потреб бізнесу, аналізі бізнес-процесів та розробці стратегій для оптимізації робочих процесів.

Цей процес допомагає організації виявити і вирішити проблеми, що виникають у її діяльності, а також впровадити нові ідеї та покращення для підвищення ефективності та конкурентоспроможності.

Задачі бізнес-аналізу включають в себе збір, аналіз та інтерпретацію даних, виявлення потреб і вимог бізнесу, а також розробку планів дій для впровадження рекомендацій [11, 12]. Він також спрямований на виявлення можливостей для оптимізації витрат, підвищення ефективності та забезпечення конкурентоспроможності підприємства на ринку.

Мета бізнес-аналізу полягає у забезпеченні цільової орієнтації підприємства, виявленні потреб клієнтів та визначенні оптимальних стратегій для задоволення цих потреб. Він сприяє прийняттю обґрунтованих рішень, що базуються на аналізі даних та експертному розумінні бізнес-середовища, що дозволяє підприємству успішно конкурувати та досягати стабільного розвитку.

## 1.2 Бізнес аналіз на текстильному підприємстві

На текстильному підприємстві бізнес-аналіз відіграє важливу роль у забезпеченні ефективного управління ресурсами та досягненні стратегічних цілей.

З урахуванням особливостей текстильного сектора, аналіз фокусується на таких ключових аспектах [12]:

- сезонність і модні тенденції: текстильні вироби піддаються впливу сезонної змінності попиту та модних тенденцій (завдяки бізнес-аналізу встановлюються відповідності між прогнозом попиту та виробництвом, що дозволяє ефективно керувати виробництвом та запасами в різних періодах року);
- управління запасами та виробництвом: запаси сировини та готової продукції впливають на фінансову ефективність підприємства (бізнес-аналіз дозволяє оптимізувати процеси управління запасами, враховуючи специфіку текстильного виробництва та мінімізуючи втрати через надлишковість або недостачу товарів);
- якість і інновації: у світі текстильної промисловості якість продукції та інноваційність грають критичну роль (бізнес-аналіз спрямований на виявлення можливостей для поліпшення якості продукції, впровадження нових технологій та підвищення конкурентоспроможності підприємства на ринку);
- ринкова стратегія: оскільки текстильний ринок динамічний та конкурентний, бізнес-аналіз сприяє розробці ефективної стратегії, враховуючи фактори, такі як цінова політика, маркетингові стратегії та вимоги споживачів (це дозволяє підприємству адаптуватися до змін на ринку та забезпечувати стабільний розвиток).

Бізнес-аналіз на текстильному підприємстві спрямований на ефективне управління виробництвом та ресурсами, забезпечення високої якості

продукції та розробку стратегій для досягнення конкурентної переваги на ринку текстилю.

### **1.3 Задачі для створення застосунку бізнес-аналізу**

Визначення цілей та потреб:

- цілі бізнесу, які потрібно досягти за допомогою програми бізнес-аналізу;
- дані, потрібні для досягнення цих цілей;
- користувачі програми та способи її використання;
- функції та можливості, необхідні користувачам.

Збір даних:

- джерела, з яких будуть збиратися дані;
- частота оновлення даних;
- формати даних, які використовуються;
- міркування щодо конфіденційності та безпеки даних.

Обробка даних:

- процес очищення та підготовки даних до аналізу;
- методи трансформації даних;
- обробка відсутніх значень та помилок.

Аналіз даних:

- типи аналізу, які будуть проводитись;
- показники та міри, які будуть використовуватись;
- візуалізація результатів аналізу.

Моделювання та прогнозування:

- необхідність створення моделей для прогнозування майбутніх тенденцій;
- типи моделей, які будуть використовуватись;
- оцінка та вдосконалення моделей.

Звітність та візуалізація:

- типи звітів та панелей моніторингу, які вам потрібні;
- методи візуалізації даних;
- способи поширення звітів та панелей моніторингу серед користувачів.

Впровадження та обслуговування:

- процес впровадження програми бізнес-аналізу на підприємстві;
- навчання користувачів користуванню програмою;
- підтримка та оновлення програми.

Безпека та захист даних:

- захист даних від несанкціонованого доступу;
- заходи безпеки для захисту даних;
- відповідність вимогам щодо конфіденційності даних, таким як GDPR та CCPA.

Інтеграція з іншими системами:

- необхідність інтеграції програми бізнес-аналізу з іншими бізнес-системами;
- API або інші методи інтеграції, які будуть використовуватись;
- забезпечення сумісності даних між різними системами.

Масштабованість:

- здатність програми бізнес-аналізу масштабуватися разом з бізнесом;
- забезпечення високої продуктивності та доступності програми;
- планування майбутнього зростання та розвитку.

#### **1.4 Аналіз популярних програм/систем для бізнес аналітики на підприємствах**

У сучасному динамічному світі бізнесу доступ до даних та їх аналіз є ключовими факторами успіху. Для цього підприємства використовують

різноманітні програми та системи бізнес-аналітики, які допомагають їм збирати, обробляти та візуалізувати дані, а також отримувати з них значущі висновки.

На ринку представлено безліч програм та систем бізнес-аналітики, кожна з яких має свої переваги та недоліки. Ось декілька популярних варіантів [16].

#### *Microsoft Power BI.*

##### Переваги:

- хмарний сервіс з широким спектром функцій;
- простий у використанні та доступний;
- популярний вибір для малого та середнього бізнесу.

##### Недоліки:

- може бути складним для складних аналітичних задач;
- деякі функції доступні лише за додаткову плату.

#### *Tableau.*

##### Переваги:

- потужна платформа з широким спектром функцій;
- складні можливості візуалізації даних;
- підходить як для початківців, так і для досвідчених аналітиків.

##### Недоліки:

- може бути дорогою для малого та середнього бізнесу;
- крива навчання може бути крутою.

#### *Qlik Sense.*

##### Переваги:

- інтуїтивно зрозумілий інтерфейс;
- самообслуговування аналітики;
- підходить для користувачів без аналітичного досвіду.

##### Недоліки:

- може бути не таким потужним, як інші платформи;
- деякі функції доступні лише за додаткову плату.

### *SAP BusinessObjects.*

#### Переваги:

- широкий спектр функцій для звітності, аналітики даних та планування;
- підходить для великих підприємств з складними потребами в аналітиці.

#### Недоліки:

- може бути складним у використанні та дорогим;
- потрібно більше ІТ-ресурсів для впровадження та обслуговування.

### *SAS.*

#### Переваги:

- потужна платформа для статистичного аналізу, машинного навчання та прогнозування;
- використовується в основному великими підприємствами та науково-дослідними установами.

#### Недоліки:

- може бути складним у використанні та дорогим;
- потрібні знання статистики та програмування.

Популярні рішення пропонують або простий підхід до використання, але обмежений функціонал, який не завжди може покрити потреби підприємства, або підхід, який вимагає додаткових знань. Окрім того, популярні рішення є платними (в повній або частковій мірі), а закритий код не дає можливості бути впевненим на 100%, що чутливі дані підприємства не опиняться в руках третьої сторони.

Розробка специфічного вузьконаправленого інструменту дозволить встановити низький поріг входу та уникнути витрат ресурсів на налаштування, а відкритий код позитивно вплине на довіру до інструменту і дозволить користувачам робити вклад в його розвиток.

## 2 АНАЛІЗ ТА ПРОЄКТУВАННЯ СИСТЕМИ

### 2.1 Аналіз вимог

#### 2.1.1 Опис цілей проєкту

Метою даного проєкту є розробка інтегрованої системи прогнозування браку на основі моделі машинного навчання для покращення управління якістю продукції.

#### 2.1.2 Функціональні вимоги

Додавання звітів:

- користувачі повинні мати можливість додавати нові звіти з даними виробничого процесу.

Обробка звітів адміністратором:

- адміністратори повинні мати можливість підтверджувати або відхиляти звіти.

Прогнозування браку:

- система повинна підтримувати прогнозування браку на основі одного набору даних;
- система повинна підтримувати прогнозування браку на основі CSV файлу з вхідними даними, з виводом у вигляді CSV файлу з додатковим стовпцем прогнозу.

Візуалізація даних:

- система повинна підтримувати візуалізацію браку конструкцій за зібраними статистичними даними.



Управління користувачами:

- адміністратор повинен мати можливість видаляти користувачів;
- адміністратор повинен мати можливість керувати доступом користувачів до ресурсів.

Перегляд системних логів:

- система повинна забезпечити можливість перегляду логів для моніторингу роботи системи.

### **2.1.3 Нефункціональні вимоги**

Продуктивність:

- система повинна забезпечувати високу швидкість обробки запитів та прогнозування.

Надійність:

- система повинна бути надійною і забезпечувати безперебійну роботу.

Безпека:

- система повинна забезпечувати захист даних користувачів та конфіденційність інформації;
- повинні бути реалізовані механізми аутентифікації та авторизації користувачів.

Масштабованість:

- система повинна бути здатна обробляти великі обсяги даних та легко масштабуватися при збільшенні кількості користувачів.

Зручність використання:

- вебінтерфейс повинен бути зручним та інтуїтивно зрозумілим.

### **2.1.4 Цільова аудиторія**

Цільовою аудиторією даної системи є:

- адміністратори: користувачі з повними правами доступу для керування системою;
- оператори: користувачі, які додають звіти і використовують систему для прогнозування та візуалізації браку.

## **2.2 Аналіз інструментів**

### **2.2.1 Мова програмування Python**

В контексті даного проєкту вибір мови Python є раціональним з наступних причин:

- чіткий та лаконічний синтаксис що сприяє швидкості розробки та масштабованості;
- використання однієї мови для розробки моделі машинного навчання та серверної частини;
- легкість інтеграції моделі машинного навчання через спільну екосистему з серверною частиною;
- багата екосистема з великою кількістю готових реалізацій типових рішень які були необхідні в даному проєкті.

### **2.2.2 Фреймворк Flask**

В контексті розробки серверної частини на Python вибір фреймворку Flask був зумовлений наступними причинами [4]:

- висока швидкість розробки;

- широка екосистема плагінів що дозволяє покрити всі потреби проєкту.

### **2.2.3 Реляційні СУБД**

Використання реляційних СУБД було обумовлене наступними факторами [13]:

- наявність чіткої структури даних для проведення операцій над ними;
- висока ефективність в порівнянні з альтернативними рішеннями;
- легкість інтеграції з Flask.

### **2.2.4 Мова програмування TypeScript**

Мова програмування TypeScript була використана для реалізації клієнтської частини з наступних причин [3]:

- наявність статичної типізації дозволяє уникнути багатьох помилок на етапі розробки;
- явне визначення типів дозволяє спростити процес розробки;
- зворотна сумісність з мовою JavaScript дозволяє використовувати її потужну екосистему;
- легкість інтеграції з React.

### **2.2.5 Фреймворк React**

Використання фреймворку React в даному проєкті обумовлене наступними причинами [1]:

- наявність широкої екосистеми;
- гарна інтеграція з TypeScript;
- потреба в розробці інтерфейсів з високим рівнем інтерактивності;
- легкість в масштабуванні і налагодженні.

### 2.3 Архітектурне проєктування

Застосунок представлений клієнт-серверною архітектурою. Серверна частина реалізована засобами мови програмування Python та фреймворку Flask. Клієнтська частина реалізована засобами мови програмування TypeScript та фреймворку React.

Діаграма, що відображає архітектуру застосунку, зображена на рисунку 2.1.

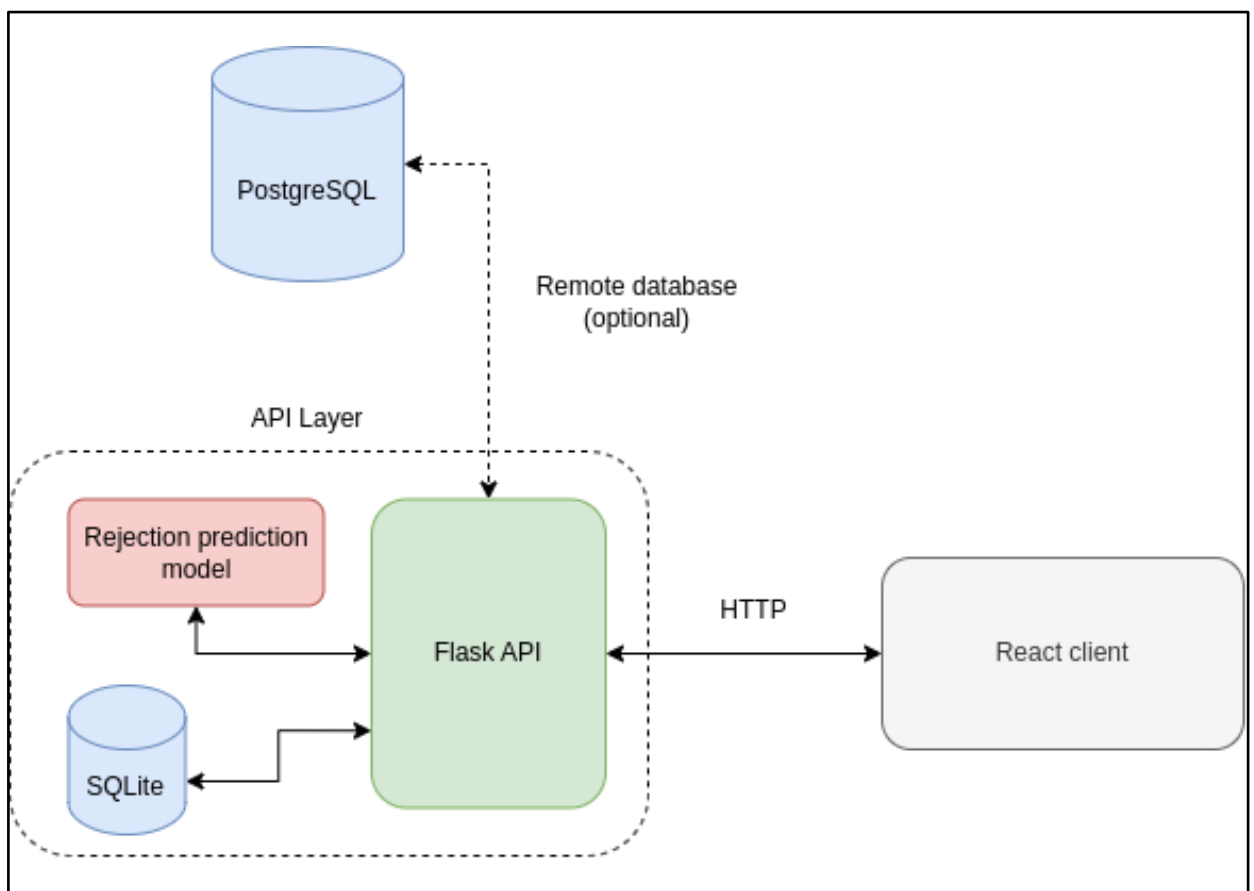


Рисунок 2.1 – Діаграма архітектури застосунку

Є опція використання віддаленої бази даних (PostgreSQL) але за потреби можна використовувати вбудовану на базі SQLite, взаємодія між клієнтом і сервером реалізується протоколом HTTP.

## 2.4 Проєктування бази даних

Для потреб застосунку було використано два види реляційних СУБД: SQLite як вбудоване рішення через його простоту та швидкість та PostgreSQL для випадків коли декілька вузлів працюють з одними даними.

Діаграма що відображає структуру бази даних зображена на рисунку 2.2.

Таблиця `users` містить всі необхідні дані для ідентифікації того чи іншого користувача в системі. Вона містить первинний ключ `id` що представлений цілим числом, символічні поля `username`, `role`, `email` та `password`.

Таблиця `logs` містить дані про системні події. Вона містить первинний ключ `id` що представлений цілим числом, цілочисельне поле `user_id` яке використовується для зв'язку з сутністю користувача який спровокував цей лог та `timestamp` типу `datetime` яке вказує на час коли лог було додано. Таблиця `logs` містить зв'язок багато до одного з таблицею `users` оскільки користувач може залишити безліч логів але один лог відноситься до одного конкретного користувача.

Таблиця `production_report` слугує буферною таблицею для збору звітів з виробництва. Вона містить первинний ключ `id` що представлений цілим числом. Поле `user_id` представлено цілим числом і вказує на сутність користувача який залишив звіт. Решта полів відповідають статистичним даним і представлені цілими числами або числами з плаваючою комою.

Таблиця `production_report` містить зв'язок багато до одного з таблицею `users` оскільки користувач може залишити безліч звітів але один звіт відноситься до одного конкретного користувача.



Рисунок 2.2 – Діаграма структури та зв'язків бази даних

Таблиця `processed_production_data` містить оброблені звіти й слугує джерелом даних для майбутнього навчання моделі а також для візуалізації статистики.

## 3 РОЗРОБКА СИСТЕМИ

### 3.1 Розробка серверної частини

#### 3.1.1 Автентифікація та авторизація

За функціонал автентифікації відповідають два ресурси:

- /register для створення нового користувача;
- /login для отримання токена доступу.

Токен доступу представлений технологією JWT, поточна реалізація зберігає в тілі токена два поля [14]:

- id: ідентифікатор користувача в системі;
- role: роль користувача в системі.

Автентифікація виконується шляхом перевірки наявності токена а авторизація шляхом перевірки ролі в тілі токена.

Паролі користувачів хешуються за допомогою алгоритму argon2 для забезпечення безпеки персональних даних [15].

#### 3.1.2 Обробка надання доступу на основі ролей

Для керування доступом на основі ролей в контексті Flask додатку було розроблено спеціальний декоратор який дозволяє визначити в єдиному місці логіку перевірки ролей, програмна реалізація зображена на рисунку 3.1.

Підхід використання декоратору були вибрано оскільки це стандартний підхід реалізації проміжної логіки в Flask додатках, використання показане на рисунку 3.2.



```

lib > decorators > role_required.py > ...
1  from functools import wraps
2  from flask_jwt_extended import get_jwt, verify_jwt_in_request
3  from flask import jsonify
4
5
6  def role_required(roles):
7      def decorator(fn):
8          @wraps(fn)
9          def wrapper(*args, **kwargs):
10             verify_jwt_in_request()
11             claims = get_jwt()
12             if claims["sub"]["role"] not in roles:
13                 return jsonify(msg="Access denied"), 403
14
15             return fn(*args, **kwargs)
16
17         return wrapper
18
19     return decorator
20

```

Рисунок 3.1 – Реалізація декоратору який реалізує перевірку доступу за роллю користувача в системі

```

17  @blp.route("/prediction")
18  class Prediction(MethodView):
19      @jwt_required()
20      @role_required(["user"])
21      @blp.arguments(PredictionInputSchema)
22      def post(self, prediction_data):
23          jwt_sub = get_jwt_identity()
24          user_id = jwt_sub["id"]
25
26          forecast = predict(prediction_data)
27          log("Get a prediction", user_id)
28          return {"result": forecast}, 200

```

Рисунок 3.2 – Приклад використання декоратору доступу на основі ролей на ресурсі для отримання прогнозу

### 3.1.3 Попередньо визначені ролі

Розглянемо попередньо призначені ролі:

- administrator: користувач з повними правами доступу, який може керувати всіма аспектами системи, включаючи користувачів, даними та конфігурацією;

- `manager`: користувач, який має права доступу для управління даними та ресурсами підприємства, але не має повних прав, як у адміністратора;
- `analyst`: користувач, який має доступ до аналітичних інструментів та звітів, що дозволяє їм аналізувати дані та робити висновки для підтримки прийняття рішень;
- `production worker`: користувач, який відповідає за зборку або інші виробничі процеси, і має обмежений доступ до системи, щоб вводити дані про виробництво.

### 3.1.4 Взаємодія з базою даних

Для організації взаємодії з базою даних використовується підхід ORM (Object-Relational Mapping). Серед переваг цього підходу – читабельність коду, простота виконання базових операцій та зручність для розробників, оскільки пропонується працювати зі знайомими абстракціями. Екосистема Flask містить розширення Flask-SQLAlchemy, яке дозволяє зручно інтегрувати SQLAlchemy в Flask без необхідності написання шаблонного коду [6].

Для початку роботи необхідно визначити метадані таблиць, зокрема, класи, що представляють сутності таблиць. Визначення класу для сутності користувача зображено на рисунку 3.3.

Після визначення відповідних таблиць ми можемо виконувати запити до них. Типові операції лаконічно і звично обробляються з використанням об'єктно-орієнтованого підходу. Операція отримання даних користувача за його ідентифікатором зображена на рисунку 3.4.

Для більш специфічних потреб SQLAlchemy надає весь функціонал, який зазвичай присутній в основних базах даних. На рисунку 3.5 показано використання просунутих функції при роботі з базою даних через SQLAlchemy.

```

5 class UserModel(db.Model, TableRepr):
6     __tablename__ = "users"
7
8     id = db.Column(db.Integer, primary_key=True)
9     username = db.Column(db.String(80), nullable=False)
10    role = db.Column(db.String(50), nullable=False)
11    email = db.Column(db.String(80), nullable=False, unique=True)
12    password = db.Column(db.String(256), nullable=False)
13
14    logs = db.relationship(
15        "LogModel", back_populates="user", lazy="dynamic", cascade="all, delete"
16    )
17    reports = db.relationship(
18        "ProductionReportModel",
19        back_populates="user",
20        lazy="dynamic",
21        cascade="all, delete",
22    )

```

Рисунок 3.3 – Визначення таблиці користувачів з використанням ORM

```

47 @blp.route("/users/me")
48 class CurrentUser(MethodView):
49     @jwt_required()
50     @blp.response(200, UserSchema)
51     def get(self):
52         jwt_sub = get_jwt_identity()
53         user_id = jwt_sub["id"]
54         user = UserModel.query.get_or_404(user_id)
55         return user
56

```

Рисунок 3.4 – Запит до таблиці users на отримання користувача за його id

```

metrca_value = (
    db.session.query(
        (
            (
                db.func.sum(ProcessedProductionDataModel.rejection)
                / db.func.sum(ProcessedProductionDataModel.total_pdn_yds)
            )
            * 100
        )
        .label("rejection_percentage")
    )
    .where(
        ProcessedProductionDataModel.warp_count
        == construction["warp_count"],
        ProcessedProductionDataModel.weft_count
        == construction["weft_count"],
        ProcessedProductionDataModel.epi == construction["epi"],
        ProcessedProductionDataModel.ppi == construction["ppi"],
    )
    .scalar()
)

```

Рисунок 3.5 – Запит на отримання значення метрик з використанням більш специфічного функціоналу баз даних

### 3.1.5 Прогнозування

Функціонал прогнозування реалізований двома ресурсами:

- /prediction – отримання прогнозу за одним набором даних;
- /prediction/bulk – отримання прогнозу по довільній кількості наборів в форматі CSV (в рамках обмежень на розмір файлу встановлених на сервері).

### 3.1.6 Збір звітності з виробництва

Збір звітності з виробництва представлений ресурсами зображеними на рисунку 3.6.



The image shows a screenshot of an API resource list. It contains five entries, each with a colored header indicating the HTTP method and a lock icon on the right. The entries are: 1. GET /reports (blue header), 2. POST /reports (green header), 3. GET /reports/{report\_id} (blue header), 4. PUT /reports/{report\_id} (orange header), and 5. DELETE /reports/{report\_id} (red header).

GET	/reports	🔒
POST	/reports	🔒
GET	/reports/{report_id}	🔒
PUT	/reports/{report_id}	🔒
DELETE	/reports/{report_id}	🔒

Рисунок 3.6 – Візуалізація ресурсів для збору звітності

Це основні операції над сутністю звіту а також ресурс для перегляду звітів з пагінацією.

### 3.1.7 Візуалізація даних

Візуалізація даних реалізована за рахунок наступних ресурсів:

- /constructions надає користувачу вибір серед доступних конструкцій (підтримує пагінацію);
- /metrics/rejection за поданими конструкціями дозволяє отримати відсоткове значення браку відштовхуючись від зібраних даних.

### 3.1.8 Адміністрування

Ресурси, пов'язані з функціоналом для адміністрування візуалізовані на рисунку 3.7:

- /admin/users/{user\_id} – операції видалення та оновлення даних користувача;
- /admin/users/{user\_id}/role – операція модифікації ролі користувача;
- /admin/logs – перегляд журналу (з підтримкою пагінації);
- /admin/reports/{report\_id}/submit – підтвердження звіту й переміщення його в кінцеву таблицю з перевіреною звітністю.

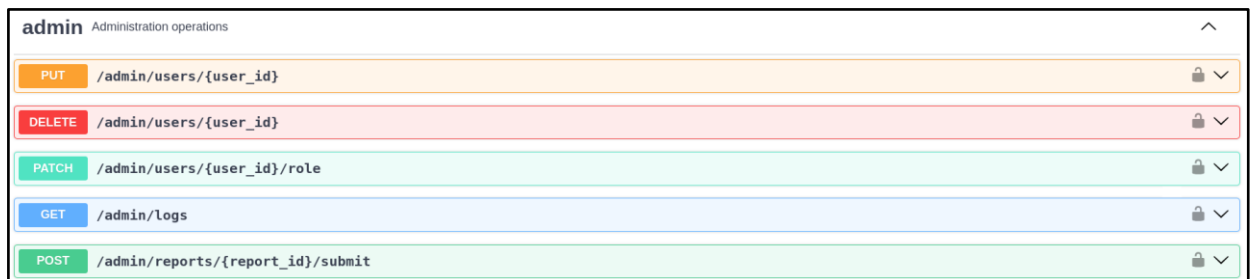


Рисунок 3.7 – Візуалізація ресурсів для адміністративних операцій

## 3.2 Розробка клієнтської частини

### 3.2.1 Загальна структура клієнту

Вебклієнт представлений SPA додатком реалізованим з використанням фреймворку React. Причиною вибору SPA було забезпечення максимально швидкого користувацького досвіду що особливо важливо на клієнтах з високим рівнем інтерактивності [9]. Для реалізації SPA підходу було використано залежність react-router [5]. Для керуванням глобальним станом додатку було використано вбудований функціонал React – Context API.

Діаграма що відображає загальну структуру клієнту (спрощено) зображена на рисунку 3.8.

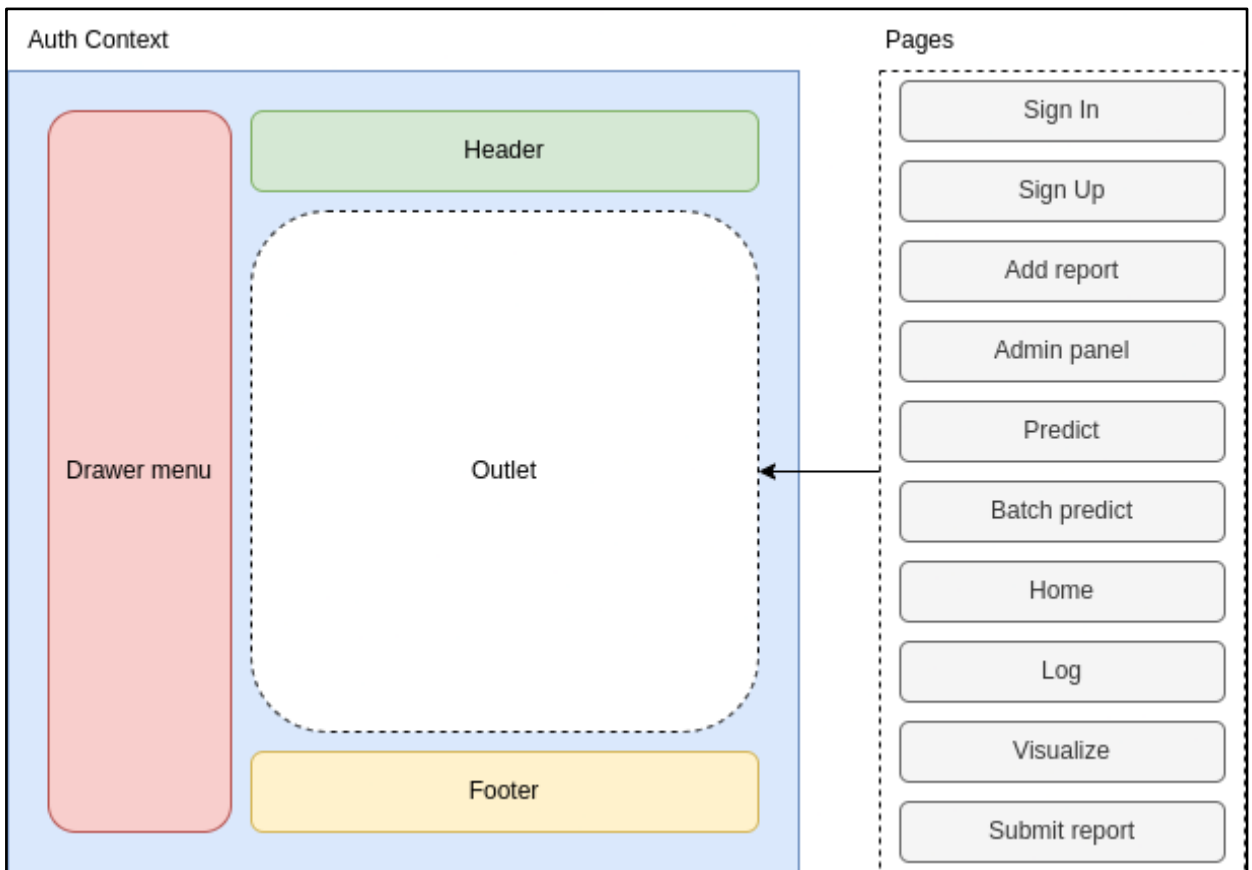


Рисунок 3.8 – Спрощена структура клієнту

Auth Context надає доступ додатку до глобального стану, він містить дані поточного користувача та токен доступу (якщо аутентифікація пройшла успішно). Вміст стану за успішної аутентифікації зображено на рисунку 3.9.

```

AuthProvider
props
  ▶ children: {$$typeof: Symbol(react.element), _owner: FiberNode...}
    new entry: ""
hooks
  1 State: {email: "user@example.com", id: 5, role: "user", us...}
    email: "user@example.com"
    id: 5
    role: "user"
    username: "user"
    new entry: ""
  2 State: "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJmcmVzaCI6ZmFsc2UsIm1hdCI

```

Рисунок 3.9 – Фактичний вміст Auth Context

Якщо за наявним в контексті токеном доступу не вдалось отримати дані користувача його буде видалено з контексту, а спроби доступу до захищених ресурсів будуть перенаправляти на сторінку логіну.

Header реалізує функціонал базової навігації. Він містить кнопку що відкриває меню, посилання на кореневу сторінку та посилання на функціонал аутентифікації. Зовнішній вигляд компоненту представлено на рисунку 3.10.



Рисунок 3.10 – Зовнішній вигляд компоненту Header

Компонент Drawer, зображений на рисунку 3.11, містить посилання на основні сторінки додатку.

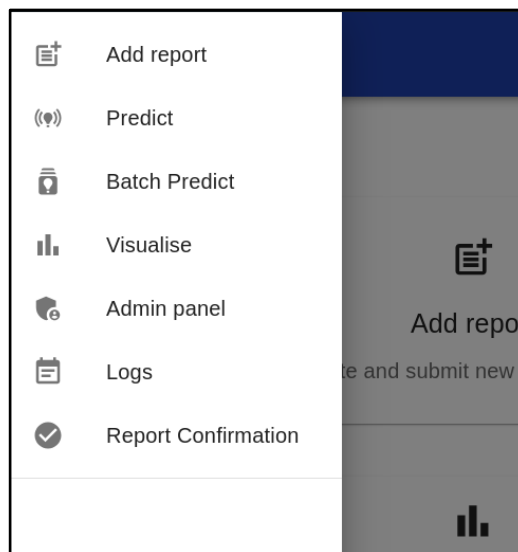


Рисунок 3.11 – Зовнішній вигляд компоненту Drawer

Footer містить назву застосунку та інформацію про ліцензію. На рисунку 3.12 зображено його поточний дизайн.

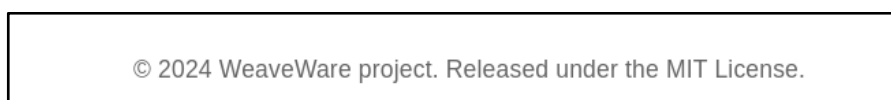
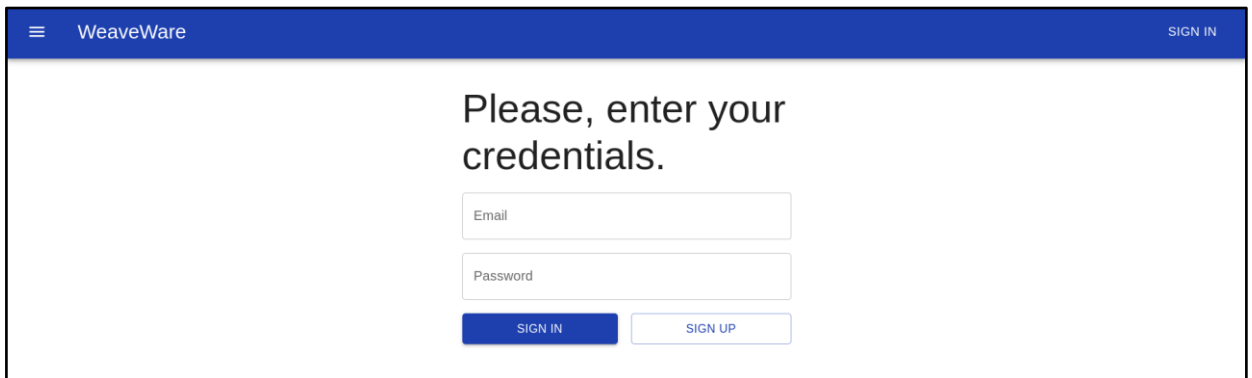


Рисунок 3.12 – Зовнішній вигляд компоненту Footer

Outlet – компонент наданий react-router що працює як слот для наших сторінок.

### 3.2.2 Опис наявних сторінок

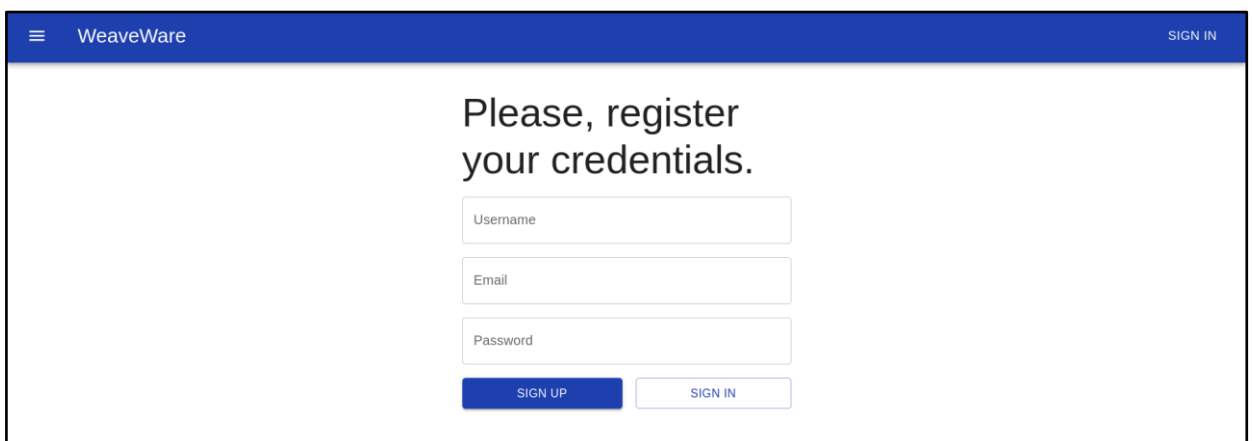
Sign In – реалізує функціонал логіну користувача, відправляє дані користувача на сервер для отримання токена доступу. Дизайн сторінки показано на рисунку 3.13.



The screenshot shows a web interface for the 'WeaveWare' application. At the top, there is a blue header bar with a hamburger menu icon on the left, the text 'WeaveWare' in the center, and a 'SIGN IN' link on the right. The main content area has a white background with the heading 'Please, enter your credentials.' centered. Below the heading are two input fields: 'Email' and 'Password'. At the bottom of the form are two buttons: a blue 'SIGN IN' button and a white 'SIGN UP' button with a blue border.

Рисунок 3.13 – Зовнішній вигляд форми логіну

Sign Up – реалізує функціонал реєстрації користувача, відправляє дані користувача на сервер для додавання в базу даних користувачів. Дизайн сторінки показано на рисунку 3.14.



The screenshot shows a web interface for the 'WeaveWare' application. At the top, there is a blue header bar with a hamburger menu icon on the left, the text 'WeaveWare' in the center, and a 'SIGN IN' link on the right. The main content area has a white background with the heading 'Please, register your credentials.' centered. Below the heading are three input fields: 'Username', 'Email', and 'Password'. At the bottom of the form are two buttons: a blue 'SIGN UP' button and a white 'SIGN IN' button with a blue border.

Рисунок 3.14 – Зовнішній вигляд форми реєстрації



Add report – надає форму для вводу даних з виробництва і відправляє їх на подальше обробку. Дизайн сторінки показано на рисунку 3.15.

The screenshot shows the 'Add report' form in the WeaveWare application. The form is titled 'Add report' and contains ten input fields arranged in two columns. Each field has a label and a value of '1'. The labels and values are as follows:

req finish fabrics	1	fabric allowance	1
rec beam length yds	1	shrink allow	1
req grey fabric	1	req beam length yds	1
total pdn yds	1	rejection	1
warp count	1	welt count	1
epl	1	ppi	1

At the bottom of the form is a blue button labeled 'SEND'. In the top right corner, there is a notification banner that says 'Report successfully added' with a green checkmark icon and a close button (X).

Рисунок 3.15 – Зовнішній вигляд форми додавання звіту

Admin panel – надає доступ до функціоналу обробки користувачів, а саме оновлення ролі та видалення користувача із системи. Дизайн сторінки показано на рисунку 3.16.

The screenshot shows the 'User Management' panel in the WeaveWare application. The panel is titled 'User Management' and contains a search box for 'User ID' and a dropdown menu for 'Role'. Below these are two buttons: 'DELETE USER' and 'CHANGE ROLE'. The 'LOG OUT' button is visible in the top right corner of the application header.

Рисунок 3.16 – Зовнішній вигляд панелі керування користувачами

Predict – надає форму для вводу вхідних даних, в подальшому робить запит на сервер для отримання прогнозу. Дизайн сторінки показано на рисунку 3.17.

Рисунок 3.17 – Зовнішній вигляд форми для отримання прогнозу за значеннями

Batch predict – надає можливість отримати прогноз за згрупованими в форматі CSV наборами вхідних даних, на вихід повертає CSV з додатковим результуючим стовпцем. Дизайн сторінки показано на рисунку 3.18.

Рисунок 3.18 – Зовнішній вигляд сторінки для отримання прогнозу за вхідним CSV файлом

Home – коренева сторінка з посиланнями та описом для основних функцій застосунку. Дизайн сторінки показано на рисунку 3.19.

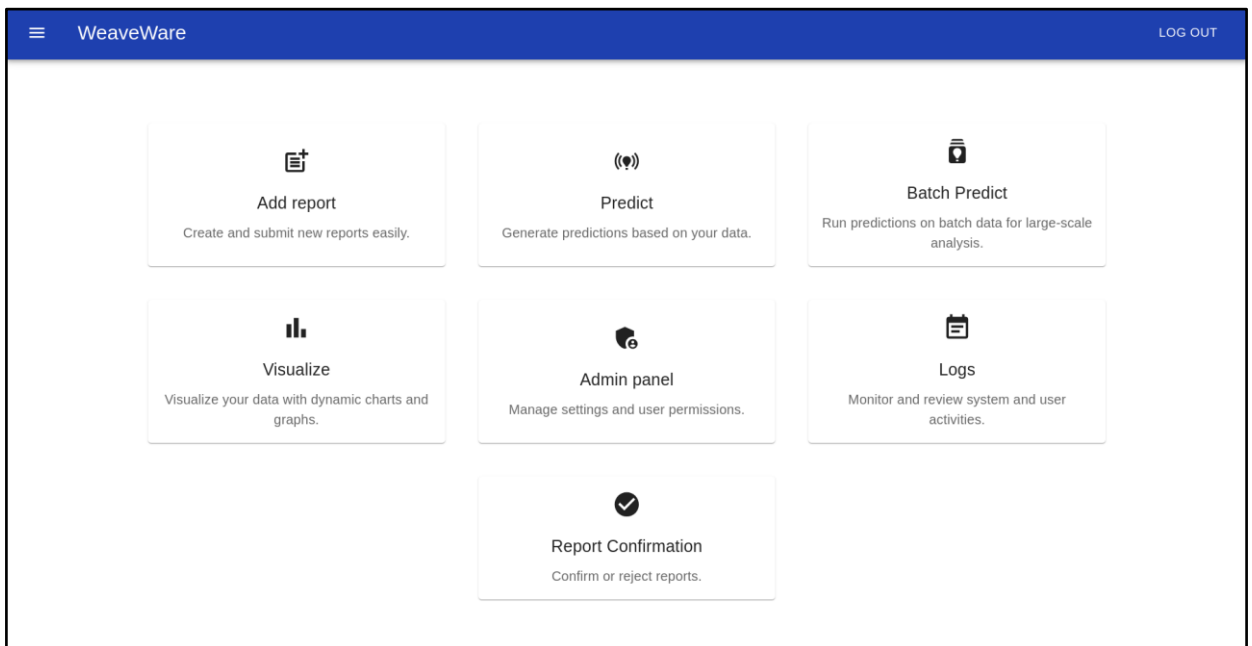


Рисунок 3.19 – Зовнішній вигляд кореневої сторінки Home

Log – сторінка що реалізує можливість перегляду системних повідомлень. Дизайн сторінки показано на рисунку 3.20.

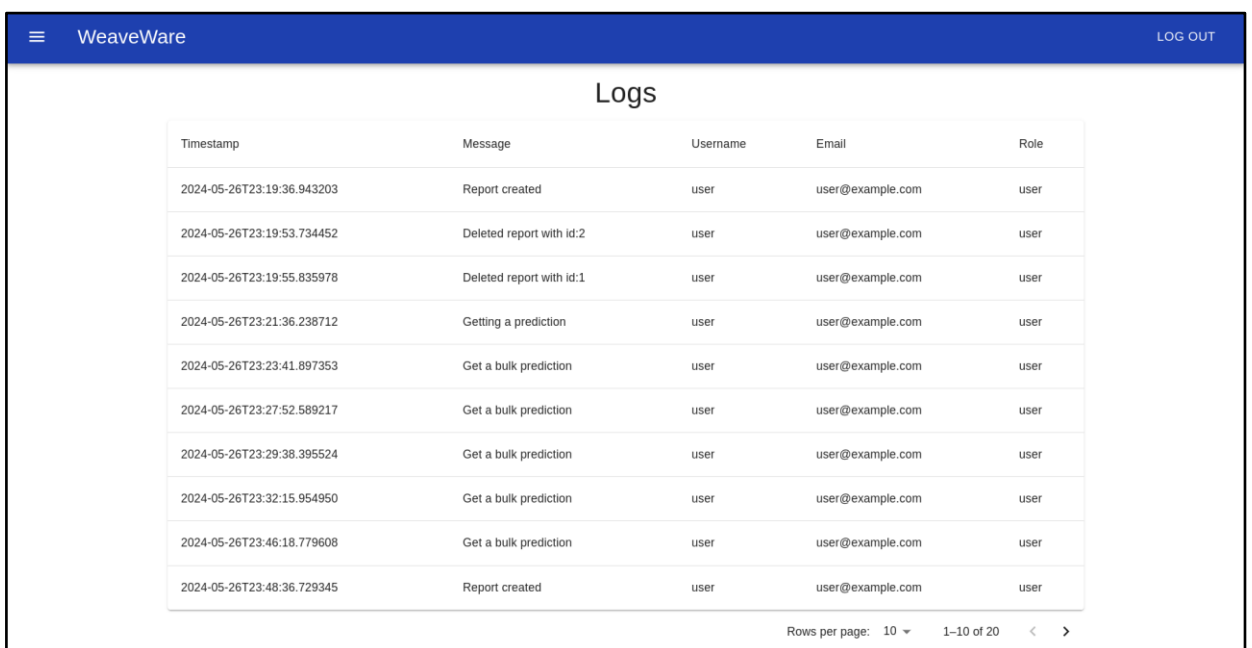


Рисунок 3.20 – Зовнішній вигляд сторінки для перегляду системних повідомлень

Visualize – сторінка що реалізує можливість візуалізації зібраних статистичних даних за конструкціями. Дизайн сторінки показано на рисунку 3.21.

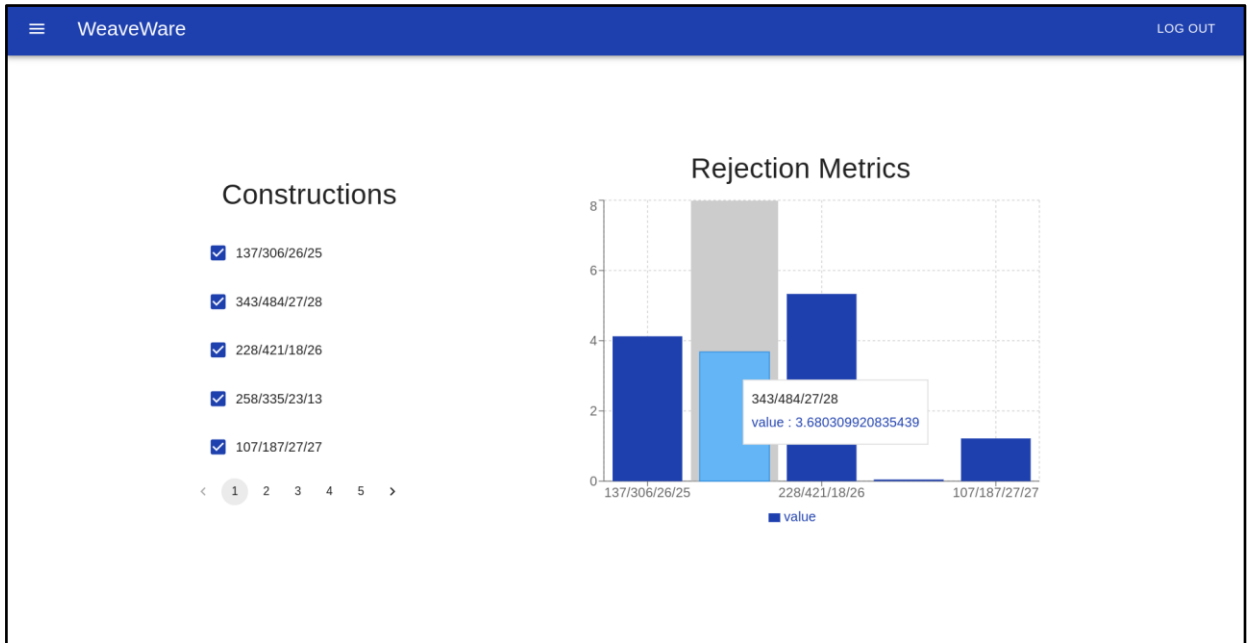


Рисунок 3.21 – Зовнішній вигляд сторінки візуалізації даних

Submit report – сторінка що реалізує можливість обробки поданих звітів. Дизайн сторінки показано на рисунку 3.22.

The screenshot displays the 'Submit report' interface in WeaveWare. The main heading is 'Report Confirmation'. Below it is a table with 10 columns: '1\_yds', 'shrink\_allow', 'req\_grey\_fabric', 'req\_beam\_length\_yds', 'total\_pdn\_yds', 'rejection', 'warp\_count', 'weft\_count', 'epi', and 'ppi'. The table contains one row with the value '1' in each of these columns. To the right of the table are two buttons: 'SUBMIT' and 'REJECT'. Below the table is a pagination control showing 'Rows per page: 10' and '1-10 of 10'.

1_yds	shrink_allow	req_grey_fabric	req_beam_length_yds	total_pdn_yds	rejection	warp_count	weft_count	epi	ppi
1	1	1	1	1	1	1	1	1	1

Рисунок 3.22 – Зовнішній вигляд сторінки обробки поданих запитів

### 3.2.3 Керування доступом до ресурсів

Для обмеження доступу до ресурсів для неавторизованих користувачів використовується спеціальний компонент-обгортка, при переході на сторінку ресурсу він перевіряє наявність токена доступу, у випадку відсутності токена користувач перенаправляється на сторінку логіну.

Реалізація компоненту-обгортки представлена на рисунку 3.23.

```
4  type Props = {
5    children: React.ReactNode;
6  };
7
8  export default function ProtectedWrapper({ children }: Props) {
9    const location = useLocation();
10   const accessToken = localStorage.getItem('accessToken');
11
12   return accessToken ? (
13     children
14   ) : (
15     <Navigate
16       to={`/${PATH.auth.group}/${PATH.auth.login}`}
17       state={{ from: location }}
18       replace
19     />
20   );
21 }
```

Рисунок 3.23 – Реалізація компонента-обгортки який захищає ресурси від доступу не авторизованих користувачів

## ВИСНОВКИ

Під час виконання проєкту було успішно реалізовано ряд завдань, які були визначені на початковому етапі роботи.

По-перше, було розглянуто бізнес-аналіз як окрему дисципліну та в контексті текстильних підприємств. Це дозволило краще зрозуміти специфіку галузі та ідентифікувати задачі, вирішення яких може принести суттєві переваги для підприємств. Такий підхід допоміг конкретизувати цільовий продукт і визначити його ключові характеристики.

По-друге, детально сформульовано наявні вимоги до продукту, що дозволило визначити, що саме, як, для чого і для кого буде реалізовано. Це було важливим етапом, який забезпечив чітке розуміння очікувань від продукту та встановив основні критерії його успішності.

Аналіз інструментів для реалізації продукту дозволив виокремити набір технологій і засобів, які, з одного боку, відповідають сформульованим вимогам, а з іншого, добре поєднуються один з одним. Такий підхід забезпечив ефективність подальшої розробки та мінімізував ризики невідповідності обраних інструментів.

На етапі проєктування було детально розглянуто та описано системи, що дозволило вирішити основні архітектурні питання до стадії реалізації. Завдяки цьому, вдалося суттєво зменшити кількість можливих помилок і хибних рішень під час безпосередньої реалізації продукту.

Етап розробки включав розробку та інтеграцію сервісів, визначених і описаних на етапі проєктування. Незважаючи на те, що багато аспектів системи було враховано на етапі проєктування, саме під час розробки було прийнято більшість ключових рішень, що визначили працездатність системи та її кінцевий вигляд.

Загалом, виконана робота дозволила розробити цілісний і ефективний продукт, який відповідає вимогам текстильних підприємств і здатний

забезпечити значні переваги в їхній діяльності. Врахування специфіки галузі, чітке формулювання вимог, ретельний вибір інструментів та детальне проєктування стали запорукою успішної реалізації проєкту.

## ПЕРЕЛІК ПОСИЛАНЬ

1. React – the library for web and native interfaces. URL: <https://react.dev/> (дата звернення: 15.01.2024).
2. Сучасний підручник з JavaScript. URL: <https://uk.javascript.info/> (дата звернення: 18.01.2024).
3. The TypeScript handbook. URL: <https://www.typescriptlang.org/docs/handbook/intro.html> (дата звернення: 27.01.2024).
4. Flask User's guide. URL: <https://flask.palletsprojects.com/en/3.0.x/> (дата звернення: 05.02.2024).
5. React Router Main Concepts. URL: <https://reactrouter.com/en/main/start/concepts> (дата звернення: 14.02.2024).
6. SQLAlchemy ORM Quickstart. URL: <https://docs.sqlalchemy.org/en/20/orm/quickstart.html> (дата звернення: 23.02.2024).
7. Flask-smorest: Flask/Marshmallow-based REST API framework. URL: <https://flask-smorest.readthedocs.io/en/latest/> (дата звернення: 04.03.2024).
8. Marshmallow: simplified object serialization. URL: <https://marshmallow.readthedocs.io/en/stable/> (дата звернення: 12.03.2024).
9. Single-page applications (SPAs) – what they are and how they work. URL: <https://business.adobe.com/blog/basics/learn-the-benefits-of-single-page-apps-spa> (дата звернення: 21.03.2024).
10. Ahmed T., Uddin S. Textile weaving dataset for machine learning to predict rejection and production of a weaving factory. *Data in Brief*. 2023. Vol. 47. P. 1–9. URL: <https://doi.org/10.1016/j.dib.2023.108995> (дата звернення: 29.03.2024).



11. Terra J. What is a Business Analysis and What does a Business Analyst Do. 2023. URL: <https://www.simplilearn.com/what-is-a-business-analyst-article> (дата звернення: 08.04.2024).
12. Malik V. How to do Business Analysis of Textile Companies. 2022. URL: <https://www.drviyajmalik.com/textile-companies/> (дата звернення: 17.04.2024).
13. Drake M. SQLite vs MySQL vs PostgreSQL: A Comparison Of Relational Database Management Systems. 2022. URL: <https://www.digitalocean.com/community/tutorials/sqlite-vs-mysql-vs-postgresql-a-comparison-of-relational-database-management-systems> (дата звернення: 26.04.2024).
14. Understanding JSON Web Tokens (JWT): A Secure Approach to Web Authentication. URL: <https://medium.com/@extio/understanding-json-web-tokens-jwt-a-secure-approach-to-web-authentication-f551e8d66deb> (дата звернення: 06.05.2024).
15. Oberai A. Demystifying password hashing algorithms. URL: <https://appwrite.io/blog/post/password-hashing-algorithms> (дата звернення: 10.05.2024).
16. The Best Data Analytics Tools Of 2024. URL: <https://www.forbes.com/advisor/business/software/best-data-analytics-tools/> (дата звернення: 10.05.2024).
17. Business intelligence: A complete overview. URL: <https://www.tableau.com/learn/articles/business-intelligence> (дата звернення: 10.05.2024).